

[illegible]

```
DDDDDDDD      TTTTTTTTTT      SSSSSSSS      PPPPPPPP      AAAAAA      RRRRRRRR      SSSSSSSS      EEEEEEEEEE
DDDDDDDD      TTTTTTTTTT      SSSSSSSS      PPPPPPPP      AAAAAA      RRRRRRRR      SSSSSSSS      EEEEEEEEEE
DD      DD      TT      SS      PP      PP      AA      AA      RR      SS      EE
DD      DD      TT      SS      PP      PP      AA      AA      RR      SS      EE
DD      DD      TT      SS      PP      PP      AA      AA      RR      SS      EE
DD      DD      TT      SSSSSS      PPPPPPPP      AA      AA      RRRRRRRR      SSSSSS      EEEEEEEE
DD      DD      TT      SSSSSS      PPPPPPPP      AA      AA      RRRRRRRR      SSSSSS      EEEEEEEE
DD      DD      TT      SS      PP      PP      AAAAAAAAAA      RR      RR      SS      EE
DD      DD      TT      SS      PP      PP      AAAAAAAAAA      RR      RR      SS      EE
DD      DD      TT      SS      PP      PP      AA      AA      RR      RR      SS      EE
DD      DD      TT      SSSSSSSS      PP      PP      AA      AA      RR      RR      SSSSSSSS      EEEEEEEEEE
DDDDDDDD      TT      SSSSSSSS      PP      PP      AA      AA      RR      RR      SSSSSSSS      EEEEEEEEEE
DDDDDDDD      TT      SSSSSSSS      PP      PP      AA      AA      RR      RR      SSSSSSSS      EEEEEEEEEE

LL      IIIIIII      SSSSSSSS
LL      IIIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL      IIIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIIII      SSSSSSSS
```

(2)	42	DECLARATIONS
(3)	60	TST\$PARSE - COMMAND PARSE ROUTINE
(4)	188	PARSE ROUTINE--PARAMETER EVALUATION AND DEFAULTING
(5)	296	PARSE ROUTINE--QUALIFIER EVALUATION
(6)	352	PARSE ROUTINE--QUALIFIER VALUE EVALUATION
(7)	549	TST\$NEXTCHAR - EXAMINE NEXT CHARACTER
(8)	630	TST\$MATCH - KEYWORD MATCH ROUTINE
(9)	688	TST\$CVTU_DTB - CONVERT UNSIGNED DECIMAL TO BINARY

```
0000 1 .TITLE TST$DTSPARSE - PARSE DTS COMMAND LINE
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27
0000 28
0000 29 :++
0000 30 : FACILITY: DTS/DTR DECNET TEST PACKAGE
0000 31
0000 32 : ABSTRACT: THIS MODULE PARSes A COMMAND LINE INPUT BY DTS.
0000 33
0000 34 : ENVIRONMENT: DTS/DTR RUN IN USER MODE AND REQUIRE NETWORK PRIVILEGE.
0000 35
0000 36 : AUTHOR: JAMES A. KRYCKA, CREATION DATE: 11-AUG-77
0000 37
0000 38 : MODIFICATIONS:
0000 39
0000 40 :--
```



```

0000 42      .SBTTL  DECLARATIONS
0000 43
0000 44 :
0000 45 : INCLUDE FILES:
0000 46 :
0000 47      FLGDEF      ; DEFINE COMMAND PARSE FLAGS
0000 48      CMDDEF      ; DEFINE COMMAND LANGUAGE SYMBOLS
0000 49      VLDDEF      ; DEFINE VALID QUALIFIER FLAGS
0000 50      .IIF NE K_LIST_MEB, .LIST MEB ; DEFINED IN DTPREFIX.MAR
0000 51 :
0000 52 : EQUATED SYMBOLS:
0000 53 :
0000 54 :      NONE
0000 55 :
0000 56 : OWN STORAGE:
0000 57 :
0000 58 :      NONE

```

```
0000 60 .SBTTL TST$PARSE - COMMAND PARSE ROUTINE
00000000 61 .PSECT TST$CODE NOWRT
0000 62 P:: ; SYMBOL FOR DEBUGGING PURPOSES
0000 63
0000 64 :++
0000 65 : FUNCTIONAL DESCRIPTION:
0000 66 :
0000 67 : NONE
0000 68 :
0000 69 : CALLING SEQUENCE:
0000 70 :
0000 71 : CALL #0,TST$PARSE
0000 72 :
0000 73 : INPUT PARAMETERS:
0000 74 :
0000 75 : R8 THE ADDRESS OF THE NEXT CHARACTER IN THE BUFFER
0000 76 : R9 THE ADDRESS OF THE END OF THE BUFFER + 1
0000 77 :
0000 78 : IMPLICIT INPUTS:
0000 79 :
0000 80 : NONE
0000 81 :
0000 82 : OUTPUT PARAMETERS:
0000 83 :
0000 84 : R0-R9 DESTROYED
0000 85 : R10 COMMAND PARAMETER VALUE (TESTTYPE)
0000 86 : R11 UPDATED PARSE FLAGS
0000 87 :
0000 88 : IMPLICIT OUTPUTS:
0000 89 :
0000 90 : TST$GB_BACK
0000 91 : TST$GB_DISPLAY
0000 92 : TST$GB_FLOW
0000 93 : TST$GB_NAK
0000 94 : TST$GT_NODENAME
0000 95 : TST$GB_PRINT
0000 96 : TST$GB_RETURN
0000 97 : TST$GB_RQUEUE
0000 98 : TST$GL_SECONDS
0000 99 : TST$GW_SIZE
0000 100 : TST$GL_SPEED
0000 101 : TST$GB_SQUEUE
0000 102 : TST$GB_TEST
0000 103 : TST$GB_TYPE
0000 104 :
0000 105 : COMPLETION CODES:
0000 106 :
0000 107 : NONE
0000 108 :
0000 109 : SIDE EFFECTS:
0000 110 :
0000 111 : NONE
0000 112 :
0000 113 : --
0000 114 :
0000 115 : .ENTRY TST$PARSE,^M<> ; ENTRY POINT
0002 116
```

```
0002 117 :  
0002 118 : DETERMINE IF THE NEXT SYNTACTICAL ELEMENT OF THE COMMAND LINE IS A  
0002 119 : PARAMETER OR QUALIFIER, OR IF THE END OF THE INPUT LINE HAS BEEN  
0002 120 : REACHED.  
0002 121 :  
0002 122 :  
02F7 30 0002 123 NEXT_ELEMENT: :  
0002 124 BSBW TST$NEXTCHAR : GET CHARACTER  
0005 125 REEXAMINE_CHAR: :  
0005 126 $CASEB SELECTOR=RO,DISPL=<- : CHARACTER:  
0005 127 END_OF_LINE- : END-OF-LINE  
0005 128 QUAC- : SLASH  
0005 129 PARSE_ERROR- : EQUAL_SIGN OR COLON  
0005 130 SPACE_OR_TAB- : SPACE OR TAB  
0005 131 PARAM- : NONE OF THE ABOVE  
0005 132 > :  
0013 133 :  
0013 134 :  
0013 135 : A SPACE OR TAB HAS BEEN ENCOUNTERED. THIS IMPLIES THAT A QUALIFIER (/)  
0013 136 : CAN NOT IMMEDIATELY FOLLOW. ANOTHER SPACE OR TAB, A PARAMETER, OR A  
0013 137 : QUALIFIER MAY FOLLOW.  
0013 138 :  
0013 139 :  
5B 08 88 0013 140 SPACE_OR_TAB: :  
EA 11 0013 141 BISB2 #FLG M DELIMITER,R11 : SET DELIMITER FLAG  
0016 142 BRB NEXT_ELEMENT : CONTINUE  
0018 143 :  
0018 144 :  
0018 145 : A QUALIFIER FOLLOWS. DISCARD THE SLASH DELIMITER AND PROCEED.  
0018 146 :  
0018 147 :  
1E 5B 03 E0 0018 148 QUAL: :  
0018 149 BBS #FLG V DELIMITER,R11,- : SLASH CANNOT FOLLOW  
001C 150 PARSE_ERROR : SPACE OR TAB  
00FD 30 001C 151 BSBW QUALIFIER : PROCESS QUALIFIER  
E4 11 001F 152 BRB REEXAMINE_CHAR : PROCESS QUALIFIER DELIMITER  
0021 153 :  
0021 154 :  
0021 155 : A PARAMETER FOLLOWS. THE COMMAND CAN HAVE ONLY ONE PARAMETER.  
0021 156 :  
0021 157 :  
15 5B 02 E2 0021 158 PARAM: :  
0021 159 BBSS #FLG V PARAMETER,R11,- : ERROR IF NOT FIRST PARAMETER  
0025 160 PARSE_ERROR :  
5B 08 8A 0025 161 BICB2 #FLG M DELIMITER,R11 : CLEAR DELIMITER FLAG  
14 10 0028 162 BSBB PARAMETER : PROCESS PARAMETER  
D9 11 002A 163 BRB REEXAMINE_CHAR : PROCESS PARAMETER DELIMITER  
002C 164 :  
002C 165 :  
002C 166 : THE END OF THE LINE HAS BEEN REACHED. SET FLAG IF THE COMMAND LINE  
002C 167 : REQUIRES ANOTHER LINE OF INPUT; OTHERWISE, DETERMINE WHETHER THE  
002C 168 : REQUIRED PARAMETER HAS BEEN RECEIVED.  
002C 169 :  
002C 170 :  
2D 51 91 002C 171 END_OF_LINE: :  
05 13 002C 172 CMPB R1,#^A/-/ : IS COMMAND LINE CONTINUED?  
002F 173 BEQLU 10$ : YES
```


L 5

16-SEP-1984 01:25:31 VAX/VMS Macro V04-00
5-SEP-1984 00:22:35 [DTS DTR.SRC]DTSPARSE.MAR:1

Page 5
(3)

Address	Disassembly	Comment
05 5B 02 E5 0031 174	BBCC	#FLG_V_PARAMETER,R11,-
0035 175		PARSE_ERROR
04 0035 176	RET	
5B 02 88 0036 177 10\$:	BISB2	#FLG_M_MULTILINE,R11
04 0039 178	RET	
003A 179		
003A 180		
003A 181		
003A 182		
003A 183		
003A 184	PARSE_ERROR:	
5B 01 88 003A 185	BISB2	#FLG_M_PARSEERROR,R11
04 003D 186	RET	


```
003E 188 .SBTTL PARSE ROUTINE--PARAMETER EVALUATION AND DEFAULTING
003E 189
003E 190 :+
003E 191 : PARAMETER IS A SPECIAL PURPOSE SUBROUTINE TO PARSE THE COMMAND PARAMETER
003E 192 : AND TO APPLY PARAMETER QUALIFIER DEFAULTS. THE PARAMETER STRING IS
003E 193 : STORED IN TST$GT_KEYWORD.
003E 194 :-
003E 195
003E 196 PARAMETER:
003E 197 MOVAL W^TST$GT_KEYWORD,R2 : CONTROL POINT
62 52 0000'CF DE 003E 197 MOVL #^A/ 7,(R2) : GET ADDRESS OF BUFFER
20202020 8F D0 0043 198 MOVL R3 : FILL KEYWORD STRING WITH SPACES
53 D4 004A 199 CLRL R3 : ZERO CHARACTER COUNT
11 11 004C 200 BRB PARAM_CHAR : STORE FIRST CHARACTER
02AB 30 004E 201 PARAM_LOOP:
004E 202 BSBW TST$NEXTCHAR : GET NEXT CHARACTER
0051 203 $CASEB SELECTOR=R0,DISPL=<- : CHARACTER:
0051 204 : PARAM_DELIMITER- : END-OF-LINE
0051 205 : PARAM_DELIMITER- : SLASH
0051 206 : PARSE_ERROR- : EQUAL_SIGN OR COLON
0051 207 : PARAM_DELIMITER- : SPACE OR TAB
0051 208 : PARAM_CHAR- : NONE OF THE ABOVE
0051 209 >
005F 210 PARAM_CHAR:
005F 211 CMPL R3,#4 : STORE ONLY FIRST 4 CHARACTERS
82 04 53 D1 005F 211 BEQLU PARAM_LOOP : IGNORE THIS CHARACTER
51 90 0062 212 MOVB R1,(R2)+ : STORE CHARACTER
53 D6 0064 213 INCL R3 : INCREMENT CHARACTER COUNT
E3 11 0067 214 BRB PARAM_LOOP : CONTINUE
006B 215 PARAM_DELIMITER:
54 0000'CF DE 006B 216 MOVAL W^TST$AZ_PARAM,R4 : GET ADDRESS OF KEYWORD TABLE
56 0000'CF DE 0070 217 MOVAL W^TST$GT_KEYWORD,R6 : GET ADDRESS OF STRING TO MATCH
02BC 30 0075 218 BSBW TST$MATCH : FIND TABLE INDEX OF KEYWORD
0000'CF 55 F6 0078 219 CVTLB R5,W^TST$GB_TEST : UPDATE TESTTYPE
5A 55 D0 007D 220 MOVL R5,R10 : SAVE IN R10 AS RETURN VALUE!!!
0080 221
0080 222
0080 223 :
0080 224 : APPLY PARAMETER QUALIFIER DEFAULTS (NOT COMMAND QUALIFIER DEFAULTS)
0080 225 : AND DENOTE VALID (PERMITTED) QUALIFIERS FOR THE COMMAND.
0080 226 :
0080 227
0080 228 $CASEB SELECTOR=R10,DISPL=<- : TEST:
0080 229 : CONNTTEST- : CONNECT TEST
0080 230 : DATATEST- : DATA TEST
0080 231 : DISCTEST- : DISCONNECT TEST
0080 232 : INTETEST- : INTERRUPT TEST
0080 233 : MISCTEST- : MISCELLANEOUS TEST
0080 234 >
008E 235 CONNTTEST:
008E 236 MOVB #DFT_K_RETURN_CO,W^TST$GB_RETURN : DEFAULTS FOR:
0000'CF 00 90 008E 236 MOVB #DFT_K_TYPE_CO,W^TST$GB_TYPE : RETURN QUALIFIER
0000'CF 01 90 0093 237 MOVB #VLD_M_NORETURN- : TYPE QUALIFIER
C8 0098 238 BISL2 : DENOTE VALID QUALIFIERS:
0099 239 :VLD_M_RETURN-
0099 240 :VLD_M_TYPE,-
0000'CF 00409000 8F 0099 241 W^TST$GL_VALID
00A1 242 RSB : EXIT
00A2 243 DATATEST:
0000'CF 00 90 00A2 243 MOVB #DFT_K_BACK,W^TST$GB_BACK : DEFAULTS FOR:
: BACK PRESSURE CONTROL
```

Address	Hex	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
---------	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------


```
011C 296 .SBTTL PARSE ROUTINE--QUALIFIER EVALUATION
011C 297
011C 298 :+
011C 299 : QUALIFIER IS A SPECIAL PURPOSE SUBROUTINE TO PARSE A COMMAND QUALIFIER OR
011C 300 : A PARAMETER QUALIFIER. THE QUALIFIER STRING IS STORED IN TST$GT_KEYWORD
011C 301 : AND THE ASSOCIATED QUALIFIER VALUE (IF ANY) IS STORED IN TST$GT_VALUE.
011C 302 :-
011C 303 QUALIFIER:
011C 304 FILLBUF DST=W^TST$GT_KEYWORD-
011C 305 SIZE=#12-
011C 306 CHAR=<#^A/ />
011C 307 MOVAL W^TST$GT_KEYWORD,R2
011C 308 CLRL R3
011C 309 QUAL_LOOP:
011C 310 BSBW TST$NEXTCHAR
011C 311 QUAL_REEXAMINE:
011C 312 $CASEB SELECTOR=R0,DISPL=<-
011C 313 QUAL_DELIMITER-
011C 314 QUAL_DELIMITER-
011C 315 QUAL_VALUE-
011C 316 QUAL_DELIMITER-
011C 317 QUAL_CHAR-
011C 318 >
011C 319 QUAL_VALUE:
011C 320 MOVAL W^TST$GT_VALUE,R2
011C 321 CLRL R3
011C 322 VALUE_LOOP:
011C 323 BSBW TST$NEXTCHAR
011C 324 $CASEB SELECTOR=R0,DISPL=<-
011C 325 QUAL_REEXAMINE-
011C 326 QUAL_REEXAMINE-
011C 327 VALUE_CHAR-
011C 328 QUAL_REEXAMINE-
011C 329 VALUE_CHAR-
011C 330 >
011C 331 VALUE_CHAR:
011C 332 CMPL R3,#8
011C 333 BEQLU VALUE_LOOP
011C 334 MOVB R1,(R2)+
011C 335 INCL R3
011C 336 BRB VALUE_LOOP
011C 337 QUAL_CHAR:
011C 338 CMPL R3,#4
011C 339 BEQLU QUAL_LOOP
011C 340 MOVB R1,(R2)+
011C 341 INCL R3
011C 342 BRB QUAL_LOOP
011C 343 QUAL_DELIMITER:
011C 344 MOVAL W^TST$AZ_QUAL,R4
011C 345 MOVAL W^TST$GT_KEYWORD,R6
011C 346 BSBW TST$MATCH
011C 347 BBS R5,W^TST$GL_VALID,108
011C 348 BRW PARSE_ERROR
011C 349 BSBB QUAL_DISPATCH
011C 350 RSB

52 0000'CF DE 0125 307
53 D4 012A 308
01CD 30 012C 309
012C 310
012F 311
012F 312
012F 313
012F 314
012F 315
012F 316
012F 317
012F 318
013D 319
52 0000'CF DE 013D 320
53 D4 0142 321
01B5 30 0144 322
0144 323
0147 324
0147 325
0147 326
0147 327
0147 328
0147 329
0147 330
0155 331
08 53 D1 0155 332
EA 13 0158 333
82 51 90 015A 334
53 D6 015D 335
E3 11 015F 336
0161 337
04 53 D1 0161 338
C6 13 0164 339
82 51 90 0166 340
53 D6 0169 341
BF 11 016B 342
016D 343
54 0000'CF DE 016D 344
56 0000'CF DE 0172 345
01BA 30 0177 346
03 0000'CF 55 E0 017A 347
FEB7 31 0180 348
01 10 0183 349
05 0185 350
```

```
.SBTTL PARSE ROUTINE--QUALIFIER VALUE EVALUATION

0186 352
0186 353
0186 354
0186 355 :+ QUAL DISPATCH IS A SPECIAL PURPOSE SUBROUTINE THAT CONTAINS QUALIFIER
0186 356 : SPECIFIC CODE. IT EXISTS AS A SUBROUTINE TO UTILIZE 'RSB' TO RETURN
0186 357 : FROM A "CASE" INSTRUCTION INSTEAD OF USING 'BRW'.
0186 358 :-
0186 359
0186 360 QUAL_DISPATCH:
56 0000'CF DE 0186 361 MOVAL W^TST$GT_VALUE,R6 : CONTROL POINT
0188 362 : GET ADDRESS OF QUALIFIER VALUE
0188 363 $CASEB SELECTOR=R5,DISPL=<- : STRING FOR POSSIBLE USE BY TST$MATCH
0188 364 BACK- : DISPATCH TO APPROPRIATE CODE
0188 365 DISPLAY- : BACK PRESSURE CONTROL
0188 366 FLOW- : DISPLAY EACH MESSAGE
0188 367 HOURS- : FLOW CONTROL
0188 368 MINUTES- : TIME OF TEST IN HOURS
0188 369 NAK- : TIME OF TEST IN MINUTES
0188 370 NOBACK- : NAK CONTROL
0188 371 NODENAME- : NO BACK PRESSURE CONTROL
0188 372 NODISPLAY- : NODENAME
0188 373 NOFLOW- : DO NOT DISPLAY EACH MESSAGE
0188 374 NONAK- : NO FLOW CONTROL
0188 375 NOPRINT- : NO NAK CONTROL
0188 376 NORETURN- : NO PRINT OPTION FOR DTR
0188 377 NOSTATISTICS- : NO USERDATA TO RETURN
0188 378 PRINT- : NO STATISTICS DESIRED
0188 379 RETURN- : PRINT OPTION FOR DTR
0188 380 RQUEUE- : RETURN USERDATA
0188 381 SECONDS- : DTR QUEUE
0188 382 SIZE- : TIME OF TEST IN SECONDS
0188 383 SPEED- : MESSAGE SIZE
0188 384 SQUEUE- : SPEED OF COMMUNICATIONS LINE
0188 385 STATISTICS- : DTS QUEUE
0188 386 TYPE- : STATISTICS DESIRED
0188 387 > : TEST TYPE (SUBFUNCTION)
0188 388 BACK: : PROCESS BACK QUALIFIER
0188 389 MOVZBL #MAX_K_BACK,R7 : DEFINE MAXIMUM VALUE
0188 390 BSBW TST$CVTU DTB : CONVERT DIGITS TO BINARY VALUE
0188 391 CVTLB R6,W^TST$GB_BACK : UPDATE BACK PRESSURE CONTROL
0188 392 RSB : EXIT
0188 393
0188 394 DISPLAY: : PROCESS DISPLAY QUALIFIER
0188 395 MOVL #MAX_K_DISPLAY,R7 : DEFINE MAXIMUM VALUE
0188 396 BSBW TST$CVTU DTB : CONVERT DIGITS TO BINARY VALUE
0188 397 CVTLB R6,W^TST$GB_DISPLAY : UPDATE DISPLAY VALUE
0188 398 RSB : EXIT
0188 399
0188 400 FLOW: : PROCESS FLOW QUALIFIER VALUE
0188 401 MOVAL W^TST$AZ_FLOW,R4 : GET ADDRESS OF KEYWORD TABLE
0188 402 BSBW TST$MATCH : FIND TABLE INDEX OF KEYWORD
0188 403 INCL R5 :
0188 404 CVTLB R5,W^TST$GB_FLOW : UPDATE FLOW CONTROL FIELD
0188 405 RSB : EXIT
0188 406
0188 407 HOURS: : PROCESS HOURS QUALIFIER VALUE
0188 408 MOVZWL #3600,-(SP) : # SECONDS IN 1 HOUR
```


57	64 8F 008A	9A 31	01EB 01EF 01F2 01F2 01F2 01F5 01FA 01FD 01FD 01FD 0201 0204 0209 020A 020A 020A 020E 020F 020F 020F 020F 0212 0214 0217 0219 0219 021E 0223 0226 0228 0229 0229 0229 022D 022E 022E 022E 0233 0233 0233 0237 0238 0238 0238 023C 023D 023D 023D	409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465	MOVZBL BRW MINUTES: MOVZBL MOVZWL BRW NAK: MOVZBL BSBW CVTLB RSB NOBACK: ASSUME CLRB RSB NODENAME: CMPL BLEQU BRW PUSHR 10\$: MOVBL MOVCL POPR RSB NODISPLAY: ASSUME CLRB RSB NOFLOW: ASSUME CLRB RSB NONAK: ASSUME CLRB RSB NOPRINT: ASSUME CLRB RSB NORETURN: ASSUME CLRB	#<MAX_K_TIME_DA/3600>,R7 ; TIME ; #60,-(SP) ; #<MAX_K_TIME_DA/60>,R7 ; TIME ; #MAX_K_NAK,R7 ; TST\$CVTU D1B ; R6,W^TST\$GB_NAK ; VAL_K_BACK_NO,EQ,0 ; W^TST\$GB_BACK ; R3,#6 ; 10\$; PARSE_ERROR ; #^M<R0,R1> ; R3,W^TST\$GT_NODENAME ; R3,W^TST\$GT_VALUE,- ; W^TST\$GT_NODENAME+1 ; #^M<R0,RT> ; VAL_K_DISP_NO,EQ,0 ; W^TST\$GB_DISPLAY ; VAL_K_FLOW_NO,EQ,0 ; W^TST\$GB_FLOW ; VAL_K_NAK_NO,EQ,0 ; W^TST\$GB_NAK ; VAL_K_PRIN_NO,EQ,0 ; W^TST\$GB_PRINT ; VAL_K_RETU_NO,EQ,0 ; W^TST\$GB_RETURN ;	; DEFINE MAXIMUM HOUR VALUE ; BRANCH TO COMMON CODE ; ; PROCESS MINUTES QUALIFIER VALUE ; # SECONDS IN 1 MINUTE ; DEFINE MAXIMUM MINUTE VALUE ; BRANCH TO COMMON CODE ; ; PROCESS NAK QUALIFIER ; DEFINE MAXIMUM VALUE ; CONVERT DIGITS TO BINARY VALUE ; UPDATE NAK CONTROL ; EXIT ; ; PROCESS NOBACK QUALIFIER ; ; UPDATE BACK PRESSURE CONTROL VALUE ; EXIT ; ; PROCESS NODENAME QUALIFIER VALUE ; A NODENAME OF 0-6 CHARACTERS ; IS ALLOWED ; IS STRING TOO LONG? ; NO, USE ENTERED VALUE ; YES ; SAVE R0 AND R1 ; NODENAME IS STORED AS A ; COUNTED ASCII STRING ; STORE LENGTH OF STRING ; STORE STRING ; NOTE R0 - R5 ARE DESTROYED! ; RESTORE R0 AND R1 ; EXIT ; ; PROCESS NODISPLAY QUALIFIER ; ; UPDATE DISPLAY VALUE ; EXIT ; ; PROCESS NOFLOW QUALIFIER ; ; UPDATE FLOW CONTROL VALUE ; EXIT ; ; PROCESS NONAK QUALIFIER ; ; UPDATE NAK CONTROL VALUE ; EXIT ; ; PROCESS NOPRINT QUALIFIER ; ; UPDATE PRINT VALUE ; EXIT ; ; PROCESS NORETURN QUALIFIER ; ; UPDATE RETURN VALUE
----	---------------	----------	--	---	--	--	---

```
05 0241 466 RSB ; EXIT
0242 467
0242 468 NOSTATISTICS: ; PROCESS NOSTATISTICS QUALIFIER
0242 469 ASSUME VAL_K_STAT_NO,ED,0 ;
0000'CF 94 0242 470 CLRB W^TST$GB_STAT ; UPDATE STATISTICS VALUE
05 0246 471 RSB ; EXIT
0247 472
0247 473 PRINT: ; PROCESS PRINT QUALIFIER
0000'CF 80 BF 90 0247 474 MOVB #VAL_K_PRIN_YES,W^TST$GB ; PRINT ; UPDATE PRINT VALUE
05 024D 475 RSB ; EXIT
024E 476
024E 477 RETURN: ; PROCESS RETURN QUALIFIER VALUE
54 0000'CF DE 024E 478 MOVAL W^TST$AZ_RETURN,R4 ; GET ADDRESS OF KEYWORD TABLE
00DE 30 0253 479 BSBW TST$MATCH ; FIND TABLE INDEX OF KEYWORD
55 D6 0256 480 INCL R5 ;
0000'CF 55 F6 0258 481 CVTLB R5,W^TST$GB_RETURN ; UPDATE RETURN USERDATA VALUE
05 025D 482 RSB ; EXIT
025E 483
025E 484 RQUEUE: ; PROCESS RQUEUE QUALIFIER VALUE
57 08 D0 025E 485 MOVL #MAX_K_RQUEUE_DA,R7 ; DEFINE MAXIMUM VALUE FOR DATA TEST
01 5A 91 0261 486 CMPB R10,#VAL_K_TEST_DATA ; IS IT A DATA TEST?
03 13 0264 487 BEQLU 10$ ; BRANCH IF YES
57 08 D0 0266 488 MOVL #MAX_K_RQUEUE_IN,R7 ; NO, DEFINE MAX VALUE FOR INT TEST
00E5 30 0269 489 10$: BSBW TST$CVTU_DTB ; CONVERT DIGITS TO BINARY VALUE
0000'CF 56 F6 026C 490 CVTLB R6,W^TST$GB_RQUEUE ; UPDATE FLOW CONTROL VALUE
05 0271 491 RSB ; EXIT
0272 492
0272 493 SECONDS: ; PROCESS SECONDS QUALIFIER VALUE
57 7E 01 D0 0272 494 MOVL #1,-(SP) ; # SECONDS IN 1 SECOND
00057E40 BF D0 0275 495 MOVL #MAX_K_TIME_DA,R7 ; DEFINE MAXIMUM SECOND VALUE
00D2 30 027C 496 TIME: BSBW TST$CVTU_DTB ; COMMON CODE
0000'CF 8E 56 C5 027F 498 MULL3 R6,(SP)+,W^TST$GL_SECONDS ; CONVERT DIGITS TO BINARY VALUE
05 0285 499 RSB ; CALCULATE NUMBER OF SECONDS
0286 500 ; EXIT
0286 501 SIZE: ; PROCESS SIZE QUALIFIER VALUE
57 1000 BF 3C 0286 502 MOVZWL #MAX_K_SIZE_DA,R7 ; DEFINE MAXIMUM VALUE FOR DATA TEST
01 5A 91 028B 503 CMPB R10,#VAL_K_TEST_DATA ; IS IT A DATA TEST?
03 13 028E 504 BEQLU 10$ ; BRANCH IF YES
57 10 D0 0290 505 MOVL #MAX_K_SIZE_IN,R7 ; NO, DEFINE MAX VALUE FOR INT TEST
00BB 30 0293 506 10$: BSBW TST$CVTU_DTB ; CONVERT DIGITS TO BINARY VALUE
0000'CF 56 F7 0296 507 CVTLW R6,W^TST$GW_SIZE ; UPDATE MESSAGE SIZE
05 029B 508 RSB ; EXIT
029C 509
029C 510 SPEED: ; PROCESS SPEED QUALIFIER VALUE
57 000F4240 BF D0 029C 511 MOVL #MAX_K_SPEED,R7 ; DEFINE MAXIMUM VALUE
00AB 30 02A3 512 BSBW TST$CVTU_DTB ; CONVERT DIGITS TO BINARY VALUE
0000'CF 56 D0 02A6 513 MOVL R6,W^TST$GL_SPEED ; UPDATE BAUD RATE
05 02AB 514 RSB ; EXIT
02AC 515
02AC 516 SQUEUE: ; PROCESS SQUEUE QUALIFIER
57 08 D0 02AC 517 MOVL #MAX_K_SQUEUE_DA,R7 ; DEFINE MAXIMUM VALUE FOR DATA TEST
01 5A 91 02AF 518 CMPB R10,#VAL_K_TEST_DATA ; IS IT A DATA TEST?
03 13 02B2 519 BEQLU 10$ ; BRANCH IF YES
57 08 D0 02B4 520 MOVL #MAX_K_SQUEUE_IN,R7 ; NO, DEFINE MAX VALUE FOR INT TEST
0097 30 02B7 521 10$: BSBW TST$CVTU_DTB ; CONVERT DIGITS TO BINARY VALUE
0000'CF 56 F6 02BA 522 CVTLB R6,W^TST$GB_SQUEUE ; UPDATE DTS QUEUE COUNT
```

```
05 02BF 523 RSB ; EXIT
02C0 524
02C0 525 STATISTICS: ; PROCESS STATISTICS QUALIFIER
02C0 526 MOVB #VAL_K_STAT_YES,W^TST$GB_STAT ; UPDATE STATISTICS VALUE
05 02C5 527 RSB ; EXIT
02C6 528
02C6 529 TYPE: ; PROCESS TYPE QUALIFIER VALUE
02C6 530 $CASEB SELECTOR=R10,DISPL=<- ; TEST:
02C6 531 10%- CONNECT TEST
02C6 532 20%- DATA TEST
02C6 533 30%- DISCONNECT TEST
02C6 534 40%- INTERRUPT TEST
02C6 535 > MISCELLANEOUS TEST BELOW
54 0000'CF DE 02D2 536 MOVAL W^TST$AZ_TYPE_MI,R4 ; GET ADDRESS OF KEYWORD TABLE
1A 11 02D7 537 BRB 50$ ; BRANCH TO COMMON CODE
54 0000'CF DE 02D9 538 10$: MOVAL W^TST$AZ_TYPE_CO,R4 ; GET ADDRESS OF KEYWORD TABLE
13 11 02DE 539 BRB 50$ ; BRANCH TO COMMON CODE
54 0000'CF DE 02E0 540 20$: MOVAL W^TST$AZ_TYPE_DA,R4 ; GET ADDRESS OF KEYWORD TABLE
0C 11 02E5 541 BRB 50$ ; BRANCH TO COMMON CODE
54 0000'CF DE 02E7 542 30$: MOVAL W^TST$AZ_TYPE_DI,R4 ; GET ADDRESS OF KEYWORD TABLE
05 11 02EC 543 BRB 50$ ; BRANCH TO COMMON CODE
54 0000'CF DE 02EE 544 40$: MOVAL W^TST$AZ_TYPE_IN,R4 ; GET ADDRESS OF KEYWORD TABLE
003E 30 02F3 545 50$: BSBW TST$MATCH ; FIND TABLE INDEX OF KEYWORD
0000'CF 55 F6 02F6 546 CVTLB R5,W^TST$GB_TYPE ; UPDATE MESSAGE TYPE
05 02FB 547 RSB ; EXIT
```

```
0000 02FC 549      .SBTTL TST$NEXTCHAR - EXAMINE NEXT CHARACTER
      02FC 550      .PSECT TST$CODE          NOWRT
      02FC 551
      02FC 552      :++
      02FC 553      : FUNCTIONAL DESCRIPTION:
      02FC 554
      02FC 555      TST$NEXTCHAR ATTEMPTS TO EXAMINE THE NEXT CHARACTER IN THE
      02FC 556      BUFFER. IF THE END OF THE BUFFER HAS BEEN REACHED, TST$NEXTCHAR
      02FC 557      SIGNALS END OF LINE CONDITION; OTHERWISE THE NEXT CHARACTER
      02FC 558      FOUND IS RETURNED ALONG WITH A VALUE INDICATING WHAT TYPE OF
      02FC 559      CHARACTER IT IS.
      02FC 560
      02FC 561      CALLING SEQUENCE:
      02FC 562
      02FC 563      BSB/JSB TST$NEXTCHAR
      02FC 564
      02FC 565      INPUT PARAMETERS:
      02FC 566
      02FC 567      R8      THE ADDRESS OF THE NEXT CHARACTER IN THE BUFFER
      02FC 568      R9      THE ADDRESS OF THE END OF THE BUFFER + 1
      02FC 569
      02FC 570      IMPLICIT INPUTS:
      02FC 571
      02FC 572      NONE
      02FC 573
      02FC 574      OUTPUT PARAMETERS:
      02FC 575
      02FC 576      R0      RESULT WHERE:
      02FC 577      0 = END OF LINE OR CHARACTER IS AN EXCLAMATION OR DASH
      02FC 578      1 = CHARACTER IS A SLASH
      02FC 579      2 = CHARACTER IS AN EQUAL SIGN OR COLON
      02FC 580      3 = CHARACTER IS A SPACE OR TAB
      02FC 581      4 = CHARACTER IS NONE OF THE ABOVE
      02FC 582      R1      THE CHARACTER EXAMINED (0 OR 'NULL' IF END OF LINE)
      02FC 583      R8      UPDATED NEXT CHARACTER POINTER
      02FC 584
      02FC 585      IMPLICIT OUTPUTS:
      02FC 586
      02FC 587      NONE
      02FC 588
      02FC 589      COMPLETION CODES:
      02FC 590
      02FC 591      NONE
      02FC 592
      02FC 593      SIDE EFFECTS:
      02FC 594
      02FC 595      NONE
      02FC 596
      02FC 597      :--
      02FC 598
      02FC 599      TST$NEXTCHAR::
      02FC 600      CLRL R0
      02FC 601      CLRL R1
      02FC 602      : ***** R0 = 0
      02FC 603      CMPL R8,R9
      02FC 604      BEQLU 10$
      02FC 605      MOVZBL (R8)+,R1
      02FC 606
      02FC 607      : CONTROL POINT
      02FC 608      : INITIALIZE RETURN VALUE
      02FC 609      : SET R1 TO 'NULL'
      02FC 610
      02FC 611      : END OF COMMAND LINE?
      02FC 612      : YES
      02FC 613      : GET NEXT CHARACTER
```

50 D4 02FC 600
51 D4 02FE 601
59 58 D1 0300 602
2E 13 0303 603
51 88 9A 0305 604
605

21	51	91	0308	606	CMPB	R1,#*A\!\	:	IS IT AN EXCLAMATION POINT?
	26	13	030B	607	BEQLU	10\$:	YES, IGNORE REST OF LINE
20	51	91	030D	608	CMPB	R1,#*A\-\	:	IS IT A DASH?
	21	13	0310	609	BEQLU	10\$:	YES, IGNORE REST OF LINE
			0312	610	:	*****	RO = 1	
	50	D6	0312	611	INCL	RO	:	INCREMENT RETURN VALUE
2F	51	91	0314	612	CMPB	R1,#*A\/\	:	IS IT A SLASH?
	1A	13	0317	613	BEQLU	10\$:	YES
			0319	614	:	*****	RO = 2	
	50	D6	0319	615	INCL	RO	:	INCREMENT RETURN VALUE
3D	51	91	031B	616	CMPB	R1,#*A\=\	:	IS IT AN EQUALS_SIGN?
	13	13	031E	617	BEQLU	10\$:	YES
3A	51	91	0320	618	CMPB	R1,#*A\:\	:	IS IT A COLON?
	0E	13	0323	619	BEQLU	10\$:	YES
			0325	620	:	*****	RO = 3	
	50	D6	0325	621	INCL	RO	:	INCREMENT RETURN VALUE
20	51	91	0327	622	CMPB	R1,#*A\ \	:	IS IT A SPACE?
	07	13	032A	623	BEQLU	10\$:	YES
09	51	91	032C	624	CMPB	R1,#*X09	:	IS IT A TAB?
	02	13	032F	625	BEQLU	10\$:	YES
			0331	626	:	*****	RO = 4	
	50	D6	0331	627	INCL	RO	:	IT'S NONE OF THE ABOVE
	05	0333	628	10\$:	RSB		:	EXIT

```
0000 0334 630 .SBTTL TST$MATCH - KEYWORD MATCH ROUTINE
      0334 631 .PSECT TST$CODE NOWRT
      0334 632
      0334 633 :++
      0334 634 : FUNCTIONAL DESCRIPTION:
      0334 635 :
      0334 636 : TST$MATCH SEARCHES THE SPECIFIED KEYWORD TABLE FOR A MATCH WITH
      0334 637 : THE SPECIFIED KEYWORD STRING. IF A MATCH IS FOUND, THE INDEX
      0334 638 : OF THE MATCHING TABLE ENTRY IS RETURNED; OTHERWISE CONTROL IS
      0334 639 : TRANSFERRED TO AN ERROR ROUTINE. THE END OF THE TABLE IS MARKED
      0334 640 : BY A NULL KEYWORD ENTRY, I.E., A NULL COUNTED ASCII STRING.
      0334 641
      0334 642 : CALLING SEQUENCE:
      0334 643 :
      0334 644 : BSB/JSB TST$MATCH
      0334 645
      0334 646 : INPUT PARAMETERS:
      0334 647 :
      0334 648 : R4 ADDRESS OF THE KEYWORD TABLE TO SEARCH
      0334 649 : R6 ADDRESS OF THE KEYWORD STRING TO MATCH AGAINST THE TABLE
      0334 650
      0334 651 : IMPLICIT INPUTS:
      0334 652 :
      0334 653 : NONE
      0334 654
      0334 655 : OUTPUT PARAMETERS:
      0334 656 :
      0334 657 : R5 THE INDEX OF THE MATCHING TABLE ENTRY IF A MATCH IS FOUND;
      0334 658 : OTHERWISE UNDEFINED.
      0334 659
      0334 660 : IMPLICIT OUTPUTS:
      0334 661 :
      0334 662 : NONE
      0334 663
      0334 664 : COMPLETION CODES:
      0334 665 :
      0334 666 : NONE
      0334 667
      0334 668 : SIDE EFFECTS:
      0334 669 :
      0334 670 : CONTROL IS TRANSFERRED TO PARSE_ERROR IF AN ERROR IS DETECTED.
      0334 671 :
      0334 672 :--
      0334 673
      0334 674 TST$MATCH::
      0334 675 PUSHR #*M<R0,R1,R2,R3,R4>
      0334 676 CLRL R5
      0334 677 10$: MOVB (R4)+,R0
      0334 678 BEQLU 30$
      0334 679 CMPC3 R0,(R4),(R6)
      0334 680 BEQLU 20$
      0334 681 ADDL3 R0,R1,R4
      0334 682 INCL R5
      0334 683 BRB 10$
      0334 684 20$: POPR #*M<R0,R1,R2,R3,R4>
      0334 685 RSB
      0334 686 30$: BRW PARSE_ERROR
```

```
1F BB 0334 675
55 D4 0334 676
50 84 0334 677
11 13 0334 678
66 64 50 29 0334 679
08 13 0334 680
54 51 50 C1 0334 681
55 D6 0334 682
ED 11 0334 683
1F BA 0334 684
FCE9 05 0334 685
31 0334 686
```

```
: CONTROL POINT
: SAVE REGISTERS
: INITIALIZE RETURN VALUE
: GET # OF SIGNIFICANT CHARACTERS
: END OF TABLE IF ZERO
: DO STRINGS MATCH?
: YES
: SET POINTER TO NEXT ENTRY
: INCREMENT TABLE INDEX
: TRY AGAIN
: RESTORE REGISTERS
: EXIT
: BRANCH TO ERROR ROUTINE
```

```
00000351 688 .SBTTL TST$CVTU_DTB - CONVERT UNSIGNED DECIMAL TO BINARY
0351 689 .PSECT TST$CODE NOWRT
0351 690
0351 691
0351 692 ++
0351 693 : FUNCTIONAL DESCRIPTION:
0351 694 : TST$CVTU_DTB CONVERTS AN UNSIGNED ASCII STRING OF 1 TO 8 DECIMAL
0351 695 : DIGITS TO A 32-BIT BINARY VALUE. IF THE RESULTANT VALUE EXCEEDS
0351 696 : THE GIVEN LIMIT, CONTROL IS TRANSFERRED TO AN ERROR ROUTINE.
0351 697
0351 698 : CALLING SEQUENCE:
0351 699 :
0351 700 : BSB/JSB TST$CVTU_DTB
0351 701
0351 702 : INPUT PARAMETERS:
0351 703 :
0351 704 : R3 LENGTH OF ASCII STRING
0351 705 : R7 MAXIMUM VALUE
0351 706
0351 707 : IMPLICIT INPUTS:
0351 708 :
0351 709 : TST$GT_VALUE = ASCII STRING TO CONVERT
0351 710
0351 711 : OUTPUT PARAMETERS:
0351 712 :
0351 713 : R2-R5 DESTROYED
0351 714 : R6 BINARY VALUE OF STRING
0351 715 : R7 UNCHANGED
0351 716
0351 717 : IMPLICIT OUTPUTS:
0351 718 :
0351 719 : NONE
0351 720
0351 721 : COMPLETION CODES:
0351 722 :
0351 723 : NONE
0351 724
0351 725 : SIDE EFFECTS:
0351 726 :
0351 727 : CONTROL IS TRANSFERRED TO PARSE_ERROR IF AN ERROR IS DETECTED.
0351 728
0351 729 :--
0351 730
0351 731 TST$CVTU_DTB::
0351 732 MOVAL W*TST$GT_VALUE,R2 : CONTROL POINT
0351 733 : GET ADDRESS OF ASCII STRING
0351 734 :
0351 735 : THE ASCII STRING IS STORED IN REVERSE ORDER, SO THE POINTER IS PLACED
0351 736 : ONE PAST THE END OF THE STRING. THEREFORE, THE STRING IS SCANNED IN
0351 737 : REVERSE ORDER TO OBTAIN THE LEAST-SIGNIFICANT TO MOST-SIGNIFICANT
0351 738 : CHARACTERS.
0351 739 :
0351 740 :
0351 741 ADDL2 R3,R2 : ADD STRING LENGTH TO POINTER
0351 742 CMPB R3,#8 : IS STRING TOO LONG?
0351 743 BGTRU 20$ : YES
0351 744 MOVZBL #1,R4 : SET-UP DIGIT PLACE VALUE
```

52 0000'CF DE

52 53 C0
08 53 91
22 1A
54 01 9A

55	56	D4	0361	745	CLRL	R6	:	ZERO RETURN VALUE
55	72	9A	0363	746	MOVZBL	-(R2),R5	:	GET NEXT ASCII CHARACTER
55	30	82	0366	747	SUBB2	#^X30,R5	:	CONVERT ASCII DIGIT TO BINARY
09	55	91	0369	748	CMPB	R5,#9	:	IS IT NON-NUMERIC?
	12	1A	036C	749	BGTRU	20\$:	YES
55	54	C4	036E	750	MULL2	R4,R5	:	MULTIPLY DIGIT BY ITS PLACE VALUE
56	55	C0	0371	751	ADDL2	R5,R6	:	ADD THIS TO THE TOTAL
54	0A	C4	0374	752	MULL2	#10,R4	:	MULTIPLY PLACE VALUE BY 10
E9	53	F5	0377	753	SOBGTR	R3,10\$:	ANOTHER DIGIT TO CONVERT?
57	56	D1	037A	754	CMPL	R6,R7	:	IS CONVERTED VALUE TOO LARGE?
	01	1A	037D	755	BGTRU	20\$:	YES
		05	037F	756	RSB		:	NO, EXIT
FCB7		31	0380	757	BRW	PARSE_ERROR	:	BRANCH TO ERROR ROUTINE
			0383	758	.END		:	

TST\$DTSPARSE
Symbol table

- PARSE DTS COMMAND LINE

L 6

16-SEP-1984 01:25:31
5-SEP-1984 00:22:35VAX/VMS Macro V04-00
[DTS\$DTR.SRC]DTSPARSE.MAR;1Page 18
(9)

```
$COUNT      = 00000004
BACK          = 000001BD R
CONNTTEST    = 0000008E R
DATATEST     = 000000A2 R
DFT_K_BACK   = 00000000
DFT_K_FLOW   = 00000002
DFT_K_NAK     = 00000000
DFT_K_RETURN_CO = 00000000
DFT_K_RETURN_DI = 00000000
DFT_K_RQUEUE_DA = 00000001
DFT_K_RQUEUE_IN = 00000001
DFT_K_SIZE_DA = 00000080
DFT_K_SIZE_IN = 00000010
DFT_K_SQUEUE_DA = 00000001
DFT_K_SQUEUE_IN = 00000001
DFT_K_TIME_DA = 0000001E
DFT_K_TIME_IN = 0000001E
DFT_K_TYPE_CO = 00000001
DFT_K_TYPE_DA = 00000000
DFT_K_TYPE_DI = 00000001
DFT_K_TYPE_IN = 00000000
DFT_K_TYPE_MI = 00000000
DISCTEST     = 000000D6 R
DISPLAY      = 000001CA R
END_OF_LINE   = 0000002C R
FLG_M_DELIMITER = 00000008
FLG_M_MULTILINE = 00000002
FLG_M_PARSERERROR = 00000001
FLG_V_DELIMITER = 00000003
FLG_V_PARAMETER = 00000002
FLOW         = 000001D6 R
HOURS        = 000001E6 R
INTETEST     = 000000EA R
K_LIST_MEB    = 00000000
MAX_K_BACK    = 00000080
MAX_K_DISPLAY = 00000026
MAX_K_NAK     = 00000080
MAX_K_RQUEUE_DA = 00000008
MAX_K_RQUEUE_IN = 00000008
MAX_K_SIZE_DA = 00001000
MAX_K_SIZE_IN = 00000010
MAX_K_SPEED   = 000F4240
MAX_K_SQUEUE_DA = 00000008
MAX_K_SQUEUE_IN = 00000008
MAX_K_TIME_DA = 00057E40
MINUTES      = 000001F2 R
MISCTEST     = 0000010D R
NAK          = 000001FD R
NEXT_ELEMENT  = 00000002 R
NOBACK       = 0000020A R
NODENAME     = 0000020F R
NODISPLAY    = 00000229 R
NOFLOW       = 0000022E R
NONAK        = 00000233 R
NOPRINT      = 00000238 R
NORETURN     = 0000023D R
NOSTATISTICS = 00000242 R
```

```
P          00000000 RG
PARAM      00000021 R
PARAMETER  0000003E R
PARAM_CHAR 0000005F R
PARAM_DELIMITER 0000006B R
PARAM_LOOP 0000004E R
PARSE_ERROR 0000003A R
PRINT      00000247 R
QUAL       00000018 R
QUALIFIER  0000011C R
QUAL_CHAR  00000161 R
QUAL_DELIMITER 0000016D R
QUAL_DISPATCH 00000186 R
QUAL_LOOP  0000012C R
QUAL_REEXAMINE 0000012F R
QUAL_VALUE 0000013D R
REEXAMINE_CHAR 00000005 R
RETURN     0000024E R
RQUEUE     0000025E R
SECONDS    00000272 R
SIZE...    = 00000001
SIZE       00000286 R
SPACE_OR_TAB 00000013 R
SPEED      0000029C R
SQUEUE     000002AC R
STATISTICS 000002C0 R
TIME       0000027C R
TST$AZ_FLOW ***** X
TST$AZ_PARAM ***** X
TST$AZ_QUAL ***** X
TST$AZ_RETURN ***** X
TST$AZ_TYPE_CO ***** X
TST$AZ_TYPE_DA ***** X
TST$AZ_TYPE_DI ***** X
TST$AZ_TYPE_IN ***** X
TST$AZ_TYPE_MI ***** X
TST$CVTU_DT 00000351 RG
TST$GB_BACK ***** X
TST$GB_DISPLAY ***** X
TST$GB_FLOW ***** X
TST$GB_NAK ***** X
TST$GB_PRINT ***** X
TST$GB_RETURN ***** X
TST$GB_RQUEUE ***** X
TST$GB_SQUEUE ***** X
TST$GB_STAT ***** X
TST$GB_TEST ***** X
TST$GB_TYPE ***** X
TST$GL_SECONDS ***** X
TST$GL_SPEED ***** X
TST$GL_VALID ***** X
TST$GT_KEYWORD ***** X
TST$GT_NODENAME ***** X
TST$GT_VALUE ***** X
TST$GW_SIZE ***** X
TST$MATCH   00000334 RG
TST$NEXTCHAR 000002FC RG
```

```
TST$PARSE    00000000 RG
TYPE         000002C6 R
VALUE_CHAR   00000155 R
VALUE_LOOP   00000144 R
VAL_K_BACK_NO = 00000000
VAL_K_DISP_NO = 00000000
VAL_K_FLOW_MESS = 00000002
VAL_K_FLOW_NO = 00000000
VAL_K_NAK_NO = 00000000
VAL_K_PRIN_NO = 00000000
VAL_K_PRIN_YES = 00000080
VAL_K_RETU_NO = 00000000
VAL_K_STAT_NO = 00000000
VAL_K_STAT_YES = 00000001
VAL_K_TEST_DATA = 00000001
VAL_K_TYPE_ABRT = 00000001
VAL_K_TYPE_ACCE = 00000001
VAL_K_TYPE_NAME = 00000000
VAL_K_TYPE_SINK = 00000000
VLD_M_BACK   = 00000001
VLD_M_FLOW   = 00000004
VLD_M_HOURS  = 00000008
VLD_M_MINUTES = 00000010
VLD_M_NAK    = 00000020
VLD_M_NOBACK = 00000040
VLD_M_NOFLOW = 00000200
VLD_M_NONAK  = 00000400
VLD_M_NORETURN = 00001000
VLD_M_RETURN = 00008000
VLD_M_RQUEUE = 00010000
VLD_M_SECONDS = 00020000
VLD_M_SIZE   = 00040000
VLD_M_SQUEUE = 00100000
VLD_M_TYPE   = 00400000
```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
TST\$CODE	00000383 (899.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	34	00:00:00.05	00:00:01.79
Command processing	118	00:00:00.60	00:00:04.84
Pass 1	215	00:00:06.14	00:00:16.75
Symbol table sort	0	00:00:00.24	00:00:00.26
Pass 2	145	00:00:02.07	00:00:05.58
Symbol table output	13	00:00:00.09	00:00:00.11
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	529	00:00:09.23	00:00:29.38

The working set limit was 1350 pages.
29598 bytes (58 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 200 non-local and 24 local symbols.
820 source lines were read in Pass 1, producing 21 object records in Pass 2.
23 pages of virtual memory were used to define 19 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-\$255\$DUA28:[DTSDTR.OBJ]DTSDTR.MLB;1	6
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	12

223 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:DTSPARSE/OBJ=OBJ\$:DTSPARSE MSRC\$:DTPREFIX/UPDATE=(ENH\$:DTPREFIX)+MSRC\$:DTSPARSE/UPDATE=(ENH\$:DTSPARSE)

0123 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

